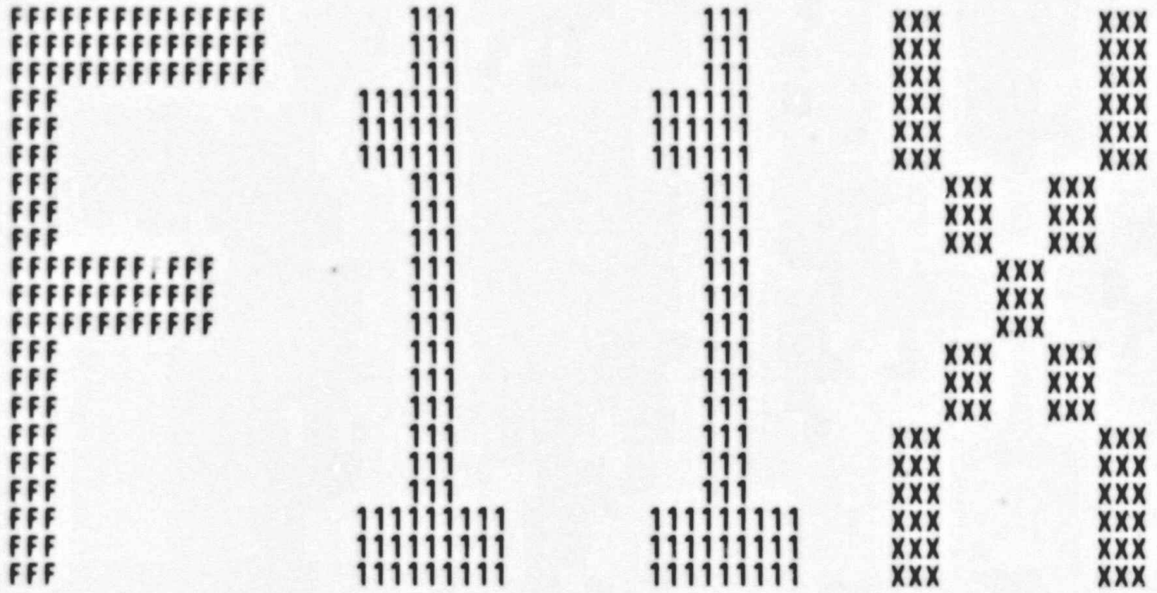

_$2

Symb

IOCS
IO_C
IO_C
IO_D
IO_F
IO_S
KICL

KILL
KILL
LB_B
LB_D
LB_F
LB_K
LB_L
LOCA
LOCA
LOCK

LOCK
LOCK
LOCK
LOC_
LOC_
L_CO
L_CO
L_DA
L_DA
MAIN
MAKE
MAKE
MAKE
MAKE
MAKE
MAKE

MAKE
MAKE
MAP_
MAP_

MAP
MARK
MARK
MARK
MARK
MARK

```
  SSSSSSSS  NN      NN  DDDDDDDD   EEEEEEEEEE  RRRRRRRR    LL
  SSSSSSSS  NN      NN  DDDDDDDD   EEEEEEEEEE  RRRRRRRR    LL
SS          NN      NN  DD      DD  EE          RR      RR  LL
SS          NN      NN  DD      DD  EE          RR      RR  LL
SS          NNNN    NN  DD      DD  EE          RR      RR  LL
SS          NNNN    NN  DD      DD  EE          RR      RR  LL
  SSSSSS    NN  NN  NN  DD      DD  EEEEEEEE    RRRRRRRR    LL
  SSSSSS    NN  NN  NN  DD      DD  EEEEEEEE    RRRRRRRR    LL
        SS  NN    NNNN  DD      DD  EE          RR  RR      LL
        SS  NN    NNNN  DD      DD  EE          RR  RR      LL
        SS  NN      NN  DD      DD  EE          RR    RR    LL              ....
        SS  NN      NN  DD      DD  EE          RR    RR    LL              ....
SSSSSSSS    NN      NN  DDDDDDDD   EEEEEEEEEE  RR      RR  LLLLLLLLLL      ....
SSSSSSSS    NN      NN  DDDDDDDD   EEEEEEEEEE  RR      RR  LLLLLLLLLL      ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
;     1          0001  0 MODULE SNDERL (
;     2          0002  0                 LANGUAGE (BLISS32),
;     3          0003  0                 IDENT = 'V04-000'
;     4          0004  0                 ) =
;     5          0005  1 BEGIN
;     6          0006  1
;     7          0007  1 !
;     8          0008  1 !**********************************************************************
;     9          0009  1 !*                                                                    *
;    10          0010  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                           *
;    11          0011  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.            *
;    12          0012  1 !*  ALL RIGHTS RESERVED.                                             *
;    13          0013  1 !*                                                                    *
;    14          0014  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
;    15          0015  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    16          0016  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER  *
;    17          0017  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
;    18          0018  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
;    19          0019  1 !*  TRANSFERRED.                                                       *
;    20          0020  1 !*                                                                    *
;    21          0021  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
;    22          0022  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
;    23          0023  1 !*  CORPORATION.                                                       *
;    24          0024  1 !*                                                                    *
;    25          0025  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
;    26          0026  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.            *
;    27          0027  1 !*                                                                    *
;    28          0028  1 !*                                                                    *
;    29          0029  1 !**********************************************************************
;    30          0030  1
;    31          0031  1 !++
;    32          0032  1 !
;    33          0033  1 ! FACILITY:  F11ACP Structure Level 1
;    34          0034  1 !
;    35          0035  1 ! ABSTRACT:
;    36          0036  1 !
;    37          0037  1 !       This routine sends a message to the error logger to inform it of a
;    38          0038  1 !       volume mount or dismount.
;    39          0039  1 !
;    40          0040  1 ! ENVIRONMENT:
;    41          0041  1 !
;    42          0042  1 !       STARLET operating system, including privileged system services
;    43          0043  1 !       and internal exec routines.
;    44          0044  1 !
;    45          0045  1 !--
;    46          0046  1 !
;    47          0047  1 !
;    48          0048  1 ! AUTHOR:  Andrew C. Goldstein,  CREATION DATE:  23-Jun-1978  18:47
;    49          0049  1 !
;    50          0050  1 ! MODIFIED BY:
;    51          0051  1 !
;    52          0052  1 !       V03-001 LMP0221         L. Mark Pilant,         27-Mar-1984  14:46
;    53          0053  1 !               Change UCB$L_OWNUIC to ORB$L_OWNER and UCB$W_VPROT to
;    54          0054  1 !               ORB$W_PROT.
;    55          0055  1 !
;    56          0056  1 !       A0101   ACG0113         Andrew C. Goldstein,    15-Jan-1980  22:58
;    57          0057  1 !               Fill in volume set data in error log message
```

```
;    58          0058  1 !
;    59          0059  1 !       A0100   ACG00001         Andrew C. Goldstein,  10-Oct-1978  20:03
;    60          0060  1 !       Previous revision history moved to F11A.REV
;    61          0061  1 !**
;    62          0062  1
;    63          0063  1
;    64          0064  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;    65          0065  1 REQUIRE 'SRC$:FCPDEF.B32';
;    66          1056  1
;    67          1057  1 !
;    68          1058  1 ! This routine is called at raised IPL and must be locked into the working set.
;    69          1059  1 !
;    70          1060  1
;    71          1061  1 LOCK_CODE;
```

```
;    73          1062  1 GLOBAL ROUTINE SEND_ERRLOG (MODE, UCB) =
;    74          1063  1
;    75          1064  1 !++
;    76          1065  1 !
;    77          1066  1 ! FUNCTIONAL DESCRIPTION:
;    78          1067  1 !
;    79          1068  1 !       This routine sends a message to the error logger to inform it of a
;    80          1069  1 !       volume mount or dismount.
;    81          1070  1 !
;    82          1071  1 !
;    83          1072  1 ! CALLING SEQUENCE:
;    84          1073  1 !       SEND_ERRLOG (ARG1, ARG2)
;    85          1074  1 !
;    86          1075  1 ! INPUT PARAMETERS:
;    87          1076  1 !       ARG1: 1 to signal mount
;    88          1077  1 !             0 to signal dismount
;    89          1078  1 !       ARG3: address of UCB
;    90          1079  1 !
;    91          1080  1 ! IMPLICIT INPUTS:
;    92          1081  1 !       NONE
;    93          1082  1 !
;    94          1083  1 ! OUTPUT PARAMETERS:
;    95          1084  1 !       NONE
;    96          1085  1 !
;    97          1086  1 ! IMPLICIT OUTPUTS:
;    98          1087  1 !       NONE
;    99          1088  1 !
;   100          1089  1 ! ROUTINE VALUE:
;   101          1090  1 !       1
;   102          1091  1 !
;   103          1092  1 ! SIDE EFFECTS:
;   104          1093  1 !       Message sent to error logger
;   105          1094  1 !
;   106          1095  1 !--
;   107          1096  1
;   108          1097  2 BEGIN
;   109          1098  2
;   110          1099  2 MAP
;   111          1100  2         UCB             : REF BBLOCK;   ! UCB argument
;   112          1101  2
;   113          1102  2 LINKAGE
;   114          1103  2         L_ERL_ALLOC     = JSB (REGISTER = 1) :
;   115          1104  2                           GLOBAL (ADDRESS = 2)
;   116          1105  2                           NOTUSED (3, 4, 5, 6, 7, 8, 9, 10, 11),
;   117          1106  2
;   118          1107  2         L_ERL_RELEASE   = JSB (REGISTER = 2) :
;   119          1108  2                           NOTUSED (3, 4, 5, 6, 7, 8, 9, 10, 11);
;   120          1109  2
;   121          1110  2 LOCAL
;   122          1111  2         ORB             : REF BBLOCK,   ! local address of ORB
;   123          1112  2         MSG_BUFFER      : REF BBLOCK;   ! other buffer pointer to dodge MOVC
;   124          1113  2
;   125          1114  2 EXTERNAL ROUTINE
;   126          1115  2         ERL$ALLOCEMB    : L_ERL_ALLOC ADDRESSING_MODE (GENERAL),
;   127          1116  2                                       ! allocate error log buffer
;   128          1117  2         ERL$RELEASEMB   : L_ERL_RELEASE ADDRESSING_MODE (GENERAL);
;   129          1118  2                                       ! release error log buffer
```

```
;  130          1119  2
;  131          1120  2
;  132          1121  2 ! Allocate an errror log buffer. If this fails, forget it.
;  133          1122  2 !
;  134          1123  2
;  135          1124  3 BEGIN
;  136          1125  3 GLOBAL REGISTER
;  137          1126  3         ADDRESS         = 2 : REF BBLOCK; ! pointer to error log buffer
;  138          1127  3
;  139          1128  3 IF NOT ERL$ALLOCEMB (EMB$K_VM_LENGTH)
;  140          1129  3 THEN RETURN 1;
;  141          1130  3 MSG_BUFFER = .ADDRESS;
;  142          1131  2 END;
;  143          1132  2
;  144          1133  2 ! Now fill in the message buffer.
;  145          1134  2 !
;  146          1135  2
;  147          1136  2 IF .MODE
;  148          1137  2 THEN MSG_BUFFER[EMB$W_VM_ENTRY] = EMB$K_VM
;  149          1138  2 ELSE MSG_BUFFER[EMB$W_VM_ENTRY] = EMB$K_VD;     ! log entry type
;  150          1139  2
;  151          1140  2 ORB = .UCB[UCB$L_ORB];
;  152          1141  2 MSG_BUFFER[EMB$L_VM_OWNUIC]     = .ORB[ORB$L_OWNER];
;  153          1142  2 MSG_BUFFER[EMB$L_VM_ERRCNT]     = .UCB[UCB$W_ERRCNT];
;  154          1143  2 MSG_BUFFER[EMB$L_VM_OPRCNT]     = .UCB[UCB$L_OPCNT];
;  155          1144  2 MSG_BUFFER[EMB$W_VM_UNIT]       = .UCB[UCB$W_UNIT];
;  156          1145  2
;  157          1146  2 MSG_BUFFER[EMB$W_VM_VOLNUM]     = 0;
;  158          1147  2 MSG_BUFFER[EMB$W_VM_NUMSET]     = 0;
;  159          1148  2
;  160          1149  2 CH$MOVE (.(BBLOCK [.UCB[UCB$L_DDB], DDB$T_NAME])<0,8> + 1,
;  161          1150  2          BBLOCK [.UCB[UCB$L_DDB], DDB$T_NAME],
;  162          1151  2          MSG_BUFFER[EMB$B_VM_NAMLNG]);
;  163          1152  2
;  164          1153  2 IF .BBLOCK[UCB[UCB$L_DEVCHAR],DEV$V_FOR]
;  165          1154  2 OR NOT .BBLOCK[UCB[UCB$L_DEVCHAR],DEV$V_SQD]
;  166          1155  2 THEN
;  167          1156  3     BEGIN
;  168          1157  3     LOCAL
;  169          1158  3         VCB             : REF BBLOCK,   ! address of volume control block
;  170          1159  3         RVT             : REF BBLOCK;   ! address of relative volume table
;  171          1160  3
;  172          1161  3     VCB = .UCB[UCB$L_VCB];
;  173          1162  3     IF .VCB[VCB$W_RVN] NEQ 0
;  174          1163  3     THEN
;  175          1164  4         BEGIN
;  176          1165  4         RVT = .VCB[VCB$L_RVT];
;  177          1166  4         MSG_BUFFER[EMB$W_VM_VOLNUM] = .VCB[VCB$W_RVN];
;  178          1167  4         MSG_BUFFER[EMB$W_VM_NUMSET] = .RVT[RVT$B_NVOLS];
;  179          1168  3         END;
;  180          1169  3     CH$MOVE (VCB$S_VOLNAME,
;  181          1170  3          BBLOCK [.UCB[UCB$L_VCB], VCB$T_VOLNAME],
;  182          1171  3          MSG_BUFFER[EMB$T_VM_LABEL]);
;  183          1172  3     END
;  184          1173  2 ELSE
;  185          1174  3     BEGIN
;  186          1175  3     LOCAL
```

```
;  187          1176  3          MVL                 : REF BBLOCK,           ! magtape volume labels
;  188          1177  3          MVL_ENTRY           : REF BBLOCK,           ! address of label entry
;  189          1178  3          RUN,                                        ! relative unit number
;  190          1179  3          RVT                 : REF BBLOCK,           ! relative volume table
;  191          1180  3          UCBLIST             : REF VECTOR,           ! address of UCB list
;  192          1181  3          VCB                 : REF BBLOCK;           ! volume control block
;  193          1182  3      VCB = .UCB[UCB$L_VCB];
;  194          1183  3      RVT = .VCB[VCB$L_RVT];
;  195          1184  3      UCBLIST = RVT[RVT$L_UCBLST];
;  196          1185  3      MVL = .VCB[VCB$L_MVL];
;  197          1186  3      MSG_BUFFER[EMB$W_VM_NUMSET] = .MVL[MVL$B_NVOLS];    ! no of volumes in vol set known
;  198          1187  3      CH$FILL(' ',VCB$S_VOLNAME,MSG_BUFFER[EMB$T_VM_LABEL]);
;  199          1188  3      INCR I FROM 0 TO .RVT[RVT$B_NVOLS] - 1 DO
;  200          1189  4          BEGIN
;  201          1190  4          RUN = .I;
;  202          1191  4          IF .UCBLIST[.I] EQL .UCB THEN EXITLOOP;
;  203          1192  3          END;
;  204          1193  3      MVL_ENTRY = .MVL + MVL$K_FIXLEN;
;  205          1194  3      INCR I FROM 0 TO .MVL[MVL$B_NVOLS] -1 DO
;  206          1195  4          BEGIN
;  207          1196  4          IF .MVL_ENTRY[MVL$B_RVN] EQL .RUN
;  208          1197  4          AND .MVL_ENTRY[MVL$V_MOUNTED]
;  209          1198  4          THEN
;  210          1199  5              BEGIN
;  211          1200  5              MSG_BUFFER[EMB$W_VM_VOLNUM] = .I + 1;
;  212          1201  5              CH$COPY(MVL$S_VOLLBL,MVL_ENTRY[MVL$T_VOLLBL],' ',
;  213          1202  5                  VCB$S_VOLNAME,MSG_BUFFER[EMB$T_VM_LABEL]);
;  214          1203  5              EXITLOOP;
;  215          1204  4              END;
;  216          1205  4          MVL_ENTRY = .MVL_ENTRY + MVL$K_LENGTH;
;  217          1206  3          END;
;  218          1207  2      END;
;  219          1208  2
;  220          1209  2  ! Finally release the buffer and make the entry.
;  221          1210  2  !
;  222          1211  2
;  223          1212  2  ERL$RELEASEMB (.MSG_BUFFER);
;  224          1213  2
;  225          1214  2  RETURN 1;
;  226          1215  2
;  227          1216  1  END;                                  ! end of routine SEND_ERRLOG


                                                    .TITLE  SNDERL
                                                    .IDENT  \V04-000\

                                                    .EXTRN  ERL$ALLOCEMB, ERL$RELEASEMB

                                                    .PSECT  $LOCKEDC1$,NOWRT,2

                                        07FC 00000  .ENTRY  SEND_ERRLOG, Save R2,R3,R4,R5,R6,R7,R8,R9,- ; 1062
                                                            R10                                         ;
                                    51    3E  D0 00002      MOVL    #62, R1                           ; 1128
                                00000000G 00  16 00005      JSB     ERL$ALLOCEMB                      ;
                                    03    50  E8 0000B      BLBS    R0, 1$                            ;
                                        00F4  31 0000E      BRW     14$                               ;
                                    59    52  D0 00011 1$:  MOVL    ADDRESS, MSG_BUFFER               ; 1130
```

```
                     07     04  AC  E9 00014        BLBC    MODE, 2$                                        ; 1136
               04    A9     40  8F  9B 00018        MOVZBW  #64, 4(MSG_BUFFER)                              ; 1137
                            05  11     0001D        BRB     3$
               04    A9     41  8F  9B 0001F 2$:    MOVZBW  #65, 4(MSG_BUFFER)                              ; 1138
                     50     08  AC  D0 00024 3$:    MOVL    UCB, R0                                         ; 1140
                     50     1C  A0  D0 00028        MOVL    28(R0), ORB
               10    A9     60  D0     0002C        MOVL    (ORB), 16(MSG_BUFFER)                           ; 1141
                     50     08  AC  D0 00030        MOVL    UCB, R0                                         ; 1142
               14    A9   0082  C0  3C 00034        MOVZWL  130(R0), 20(MSG_BUFFER)
                     50     08  AC  D0 0003A        MOVL    UCB, R0                                         ; 1143
               18    A9     70  A0  D0 0003E        MOVL    112(R0), 24(MSG_BUFFER)
                     50     08  AC  D0 00043        MOVL    UCB, R0                                         ; 1144
               1C    A9     54  A0  B0 00047        MOVW    84(R0), 28(MSG_BUFFER)
                            2E  A9  D4 0004C        CLRL    46(MSG_BUFFER)                                  ; 1146
                     50     08  AC  D0 0004F        MOVL    UCB, R0                                         ; 1149
                     50     28  A0  D0 00053        MOVL    40(R0), R0
                     51     14  A0  9A 00057        MOVZBL  20(R0), R1
                                51  D6 0005B        INCL    R1
       1E    A9      14     A0  51  28 0005D        MOVC3   R1, 20(R0), 30(MSG_BUFFER)                      ; 1151
                     51     08  AC  D0 00063        MOVL    UCB, R1                                         ; 1161
                     50     08  AC  D0 00067        MOVL    UCB, R0                                         ; 1153
                     05     3B  A0  E8 0006B        BLBS    59(R0), 4$
             27      38     A0  05  E0 0006F        BBS     #5, 56(R0), 6$                                  ; 1154
                     50     34  A1  D0 00074 4$:    MOVL    52(R1), VCB                                     ; 1161
                            0E  A0  B5 00078        TSTW    14(VCB)                                         ; 1162
                            0E  13     0007B        BEQL    5$
                     51     20  A0  D0 0007D        MOVL    32(VCB), RVT                                    ; 1165
               2E    A9     0E  A0  B0 00081        MOVW    14(VCB), 46(MSG_BUFFER)                         ; 1166
               30    A9     0B  A1  9B 00086        MOVZBW  11(RVT), 48(MSG_BUFFER)                         ; 1167
                     50     08  AC  D0 0008B 5$:    MOVL    UCB, R0                                         ; 1170
                     50     34  A0  D0 0008F        MOVL    52(R0), R0
       32    A9      14     A0  0C  28 00093        MOVC3   #12, 20(R0), 50(MSG_BUFFER)                     ; 1171
                            61  11     00099        BRB     13$                                             ; 1153
                     50     34  A1  D0 0009B 6$:    MOVL    52(R1), VCB                                     ; 1182
                     56     20  A0  D0 0009F        MOVL    32(VCB), RVT                                    ; 1183
                     58     44  A6  9E 000A3        MOVAB   68(R6), UCBLIST                                 ; 1184
                     57     34  A0  D0 000A7        MOVL    52(VCB), MVL                                    ; 1185
               30    A9     0B  A7  9B 000AB        MOVZBW  11(MVL), 48(MSG_BUFFER)                         ; 1186
 0C          20      6E     00  2C     000B0        MOVC5   #0, (SP), #32, #12, 50(MSG_BUFFER)              ; 1187
                            32  A9     000B5
                     51     0B  A6  9A 000B7        MOVZBL  11(RVT), R1                                     ; 1188
                     50     01  CE     000BB        MNEGL   #1, I                                           ; 1191
                            0A  11     000BE        BRB     8$
                     5A     50  D0     000C0 7$:    MOVL    I, RUN                                          ; 1190
               08    AC   6840  D1     000C3        CMPL    (UCBLIST)[I], UCB                               ; 1191
                            04  13     000C8        BEQL    9$
             F2      50     51  F2     000CA 8$:    AOBLSS  R1, I, 7$                                       ; 1188
                     58     24  A7  9E 000CE 9$:    MOVAB   36(R7), MVL_ENTRY                               ; 1193
                     57     0B  A7  9A 000D2        MOVZBL  11(MVL), R7                                     ; 1194
                     56     01  CE     000D6        MNEGL   #1, I
                            1D  11     000D9        BRB     12$
 5A    06    A8      08     00  ED     000DB 10$:   CMPZV   #0, #8, 6(MVL_ENTRY), RUN                       ; 1196
                            12  12     000E1        BNEQ    11$
                     0E     07  A8  E9 000E3        BLBC    7(MVL_ENTRY), 11$                               ; 1197
       2E    A9      56     01  A1     000E7        ADDW3   #1, I, 46(MSG_BUFFER)                           ; 1200
 0C          20      68     06  2C     000EC        MOVC5   #6, (MVL_ENTRY), #32, #12, 50(MSG_BUFFER)       ; 1202
                            32  A9     000F1
```

```
                                    07  11 000F3         BRB     13$                                          ; 1199
                              58    08  C0 000F5 11$:    ADDL2   #8, MVL_ENTRY                                ; 1205
               DF             56    57  F2 000F8 12$:    AOBLSS  R7, I, 10$                                   ; 1194
                              52    59  D0 000FC 13$:    MOVL    MSG_BUFFER, R2                               ; 1212
                        00000000G   00  16 000FF         JSB     ERL$RELEASEMB                                ;
                              50    01  D0 00105 14$:    MOVL    #1, R0                                       ; 1214
                                        04 00108         RET                                                  ; 1216

; Routine Size:  265 bytes,    Routine Base:  $LOCKEDC1$ + 0000


;  228          1217  1
;  229          1218  1 END
;  230          1219  0 ELUDOM




;                               PSECT SUMMARY
;
;       Name                          Bytes                       Attributes
;
;   $LOCKEDC1$                          265  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)




;                         Library Statistics
;
;                                               -------- Symbols --------         Pages         Processing
;       File                                    Total    Loaded   Percent         Mapped        Time
;
;   _$255$DUA28:[SYSLIB]LIB.L32;1               18619        54         0         1000            00:02.0




;                                COMMAND QUALIFIERS
;
;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:SNDERL/OBJ=OBJ$:SNDERL MSRC$:SNDERL/UPDATE=(ENH$:SNDERL)

; Size:             265 code + 0 data bytes
; Run Time:            00:12.6
; Elapsed Time:        00:25.5
; Lines/CPU Min:        5795
; Lexemes/CPU-Min: 26396
; Memory Used:  178 pages
; Compilation Complete
```

SCHFCB
LIS

SNDSMB
LIS

SHFDIR
LIS

SNDERR
LIS

TRUNC
LIS

FAL

FAL
MAP

SELVOL
LIS

DAPDEF
MDL

SMALOC
LIS

SNDBAD
LIS

SWITVL
LIS

WITURN
LIS